



## L'avenir du web au prisme de la ressource

Nicolas Delaforge, Fabien Gandon, Alexandre Monnin

### ► To cite this version:

Nicolas Delaforge, Fabien Gandon, Alexandre Monnin. L'avenir du web au prisme de la ressource. Lisette Calderan and Pascale Laurent and Hélène Lowinger and Jacques Millet. Le document numérique à l'heure du web, ADBS, pp.229-252, 2012, Sciences et techniques de l'information, 978-2-84365-142-7. hal-00739526v2

**HAL Id: hal-00739526**

**<https://inria.hal.science/hal-00739526v2>**

Submitted on 12 Jul 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# L'avenir du web au prisme de la ressource

Nicolas Delaforge, Fabien Gandon et Alexandre Monnin

*Ingénieur de recherche et doctorant au sein de l'équipe Wimmics, Nicolas Delaforge est en charge de la coordination scientifique et de la direction technique du projet ANR ISICIL depuis février 2009. Après un DEA en sciences cognitives à l'Université de technologie de Compiègne (UTC), il a travaillé dans différents projets innovants d'ingénierie des connaissances. Il a effectué la première partie sa thèse sous la direction de Bruno Bachimont, à l'Institut national de l'audiovisuel (INA), en participant notamment à la préfiguration de la plateforme technique pour l'archivage du web français. Après un passage chez Orange, il a rejoint l'Inria où il renoue avec les questionnements théoriques sur le statut documentaire du web et reprend ses travaux de recherche sous la direction de Fabien Gandon. nicolas.delaforge@inria.fr*

*Docteur et habilité à diriger les recherches en informatique, Fabien Gandon est responsable de l'équipe Wimmics (Inria, I3S, CNRS, Université de Nice Sophia Antipolis) et représentant de l'Institut de recherche en sciences du numérique (INRIA) au World Wide Web Consortium (W3C). Il effectue et dirige des recherches au carrefour du web sémantique, du web social et des systèmes à base de connaissances. fabien.gandon@inria.fr*

*Responsable Recherche web et métadonnées à l'Institut de recherche et d'innovation (IRI) du Centre Pompidou où il dirige plusieurs projets innovants et séminaires de recherche, Alexandre Monnin achève une thèse de philosophie à l'université Paris 1 (équipe Phico, composante EXeCO) consacrée à la philosophie du web. Discipline autour de laquelle il organise chaque année depuis 2010 les événements scientifiques « PhiloWeb », et à laquelle un numéro spécial de la revue Metaphilosophy a été récemment consacré. Il est collaborateur extérieur de l'INRIA (membre associé de l'équipe Wimmics) sur le projet SemanticPedia (incluant le DBpedia francophone) qu'il a contribué à lancer, également associé au CNAM (laboratoire DICEN), et chair et co-fondateur du « Philosophy of the Web » Community Group du W3C. <http://web-and-philosophy.org>, [aamonnz@gmail.com](mailto:aamonnz@gmail.com)*

De plus en plus fréquemment, le web s'intercale entre nous et le monde. Le web des documents et des données augmente nos perceptions de la réalité quand, dans le même temps, le web d'applications et de services accroît l'emprise que nous exerçons sur elle en multipliant les tâches que nous pouvons accomplir. Devenu incontournable dans nos activités quotidiennes, il est également difficilement gérable.

Sur le web, une ressource peut être n'importe quoi et, à mesure que le réseau s'étend, tout, dans notre environnement, est susceptible de se muer en ressource. Certes, on parle maintenant de « réalité augmentée » par le web mais, à mesure que la toile se déploie, il faut également noter à quel point la réalité vient l'augmenter elle-même, en raison de la quantité et de la diversité de ressources identifiées en son sein.

Cette question était déjà latente avec le « web documentaire » – désignation sous laquelle il fut volontiers présenté dans les années 1990. Le modèle documentaire initialement mobilisé pour en rendre compte, où chaque page HTML accédée était un document téléchargé par un programme client (le navigateur) auprès d'un programme serveur (serveur web), est rapidement apparu beaucoup trop étroit pour englober des scénarios de cas particuliers tels que les pages dynamiques, la négociation de contenu, les applications spécifiques comme un annuaire, etc. L'aspect calculatoire, qui parut constituer une exception à la métaphore de la « bibliothèque universelle du web », devint en fin de compte la règle. En outre, le web sémantique et ses métadonnées, portant potentiellement sur tout ce qui est identifiable, soulevèrent de nouvelles interrogations que l'on peut résumer à l'aide du raccourci suivant : comment est-il possible d'identifier sur le web des choses résidant en dehors du web ? L'architecture REST<sup>1</sup> suggère un élément de réponse, présent dès la naissance du web, en fournissant une grille de lecture *a posteriori* le définissant comme une « application centrée (ou orientée) ressources ».

Les évolutions récentes du web (AJAX, HTML5, Linked Data, etc.) et les évolutions envisagées pour l'avenir (web des objets, web ubiquitaire, etc.) imposent de jeter un regard neuf sur son architecture. Après examen, la ressource pourrait bien constituer le concept unificateur de toutes ses facettes et la pierre angulaire susceptible de rendre compte de sa cohérence architecturale. Il nous faut donc la redéfinir au plus proche de son évolution dans les standards, au fil du temps mais aussi des usages. De cette définition et caractérisation précise dépendra notre capacité à tirer pleinement parti des ressources en les identifiant, publiant, cherchant, filtrant, combinant, adaptant, décrivant, etc.

---

<sup>1</sup> REST (REpresentational State Transfer) est un style d'architecture logicielle qui, appliqué au web, fait que chaque ressource est nommée par une URI et a une représentation sur laquelle chaque opération HTTP (lecture, modification) est autosuffisante.

Nous proposons donc ici d'entériner l'abandon d'une analyse centrée sur une conception substantielle du document au profit d'une vision mettant en avant une toile de ressources informatiques, un graphe de ressources computationnelles liées, et ce en tirant parti des concepts élaborés par les architectes du Web eux-mêmes. La définition de la ressource proposée par Justin Erenkrantz [11], entendue comme un « *locus* de calcul » dans ses travaux consacrés à CREST<sup>2</sup> et les conséquences de la transformation des objets de notre environnement en de semblables *loci*, constitueront notre point de départ dans ce chapitre.

Il semble également que, parmi les différentes facettes du web, le web de données liées (c'est-à-dire les données liées et structurées sur le web et selon ses standards) soit appelé à jouer un rôle important dans l'avenir. Pour gérer la diversité des ressources, nous pouvons en effet lui associer une autre diversité : celle des métadonnées. Nous pensons qu'un web de descriptions sémantiques contrôlées au travers de leurs schémas est à même de doter la toile d'un paradigme distribué et extensible pour modéliser et gérer la collection indéfinie (car ouverte) de ses ressources au travers de leurs modèles.

Pour cette raison, nous allons jeter les bases théoriques de notre travail avec, comme horizon, de nous approcher de la constitution d'une ontologie des ressources basée sur les formalismes du web sémantique. Et ce afin de résoudre les nombreux problèmes afférents, trop souvent réduits à un discours portant sur les seules URIs – la nature de la ressource, question pourtant cruciale, n'étant en définitive que trop rarement interrogée.

## 1 Ressources web : tourner la page du web documentaire

### 1.1 Nommer sur le web

Dans cette première partie, nous entendons démontrer qu'il est possible, d'un point de vue strictement architectural, de rendre compte de la transition supposée d'un web du document vers un web d'applications. Nous ne proposons pas ici un compte rendu historique du développement du web interactif, assorti d'une analyse minutieuse de l'introduction de JavaScript, du DOM, des applications fondées sur Ajax, etc. Les éléments de base de la toile et les contraintes qu'ils impliquent ont en effet suffi à rendre possibles, si ce n'est à encourager ses nombreuses évolutions. Dès sa naissance le web recelait les éléments clefs de ce qui put apparaître par la suite comme des révolutions.

Au cœur de l'architecture originale du web [3], nous trouvons trois composants de base : l'identifiant (URI), le protocole (HTTP) et le langage (HTML).

Le premier concept de base est l'URL [5] ou URI [6] [8]. Au fil du temps, l'URI<sup>3</sup> fut considérée comme un format d'identificateurs uniques pour nommer n'importe quelle « ressource » sur le web. Cette conceptualisation des URIs provient du style d'architecture REST, défini par Roy Fielding au cours de la seconde moitié des années 1990 et formalisée dans sa thèse finalisée en l'an 2000 [13], puis dans un article paru en 2002, co-rédigé avec son directeur de thèse [14]. Le travail et les discussions portant sur REST ont non seulement fourni un nouveau paradigme de conception et de composition d'applications web ; ils nous ont aussi et surtout amenés à réinterpréter les notions fondamentales du web afin de répondre aux difficultés suscitées jusque-là par les standards existants.

Si une URI prescrit comment obtenir la représentation d'une ressource, elle est alors également une URL<sup>4</sup>, c'est-à-dire l'une de ces célèbres « adresses » web avec lesquelles nous sommes tous devenus familiers, même si, à l'origine, elles ne devaient nullement être manipulées directement par les utilisateurs. « <http://www.inria.fr> » est l'URI dite « déréréférençable » (permettant d'accéder ou de générer un contenu consultable) de la page d'accueil de l'INRIA.

Bien que lesdites adresses ne fussent pas initialement destinées à être utilisées par qui que ce fût, elles sont désormais parties intégrantes de nos communications quotidiennes, au point qu'il existe un véritable marché où les noms de domaine sont évalués et échangés. On croit les connaître et pourtant, davantage que des adresses, les URLs constituent en réalité un sous-ensemble des URIs qui ont la particularité d'être déréréférençables. Aussi les URLs ne servent-elles pas seulement à localiser des représentations. Elles ajoutent une relation d'accès aux représentations en plus de la relation d'identification établie entre les URIs et les ressources identifiées. Cette relation d'identification prime, au même titre que les URIs précèdent logiquement les URLs qui n'en sont qu'un sous-ensemble dérivé. Ce sous-ensemble des URIs est appelé de manière plus précise HTTP-URI, indiquant ainsi la double relation d'identification et d'accès à la représentation caractéristique de l'utilisation du protocole HTTP.

Le deuxième concept fondamental, *i.e.* le protocole HTTP [12], permet à un client (typiquement un navigateur web) de demander une représentation de la ressource identifiée et « localisée » par une URL, et d'obtenir en retour soit les données du codage de cette représentation soit un code d'erreur. HTTP propose un large panel de codes d'erreurs indiquant plus ou moins précisément la nature du problème. Par exemple, la fameuse erreur 404 indiquant que la page demandée n'a pas été trouvée. Le protocole HTTP ne permet d'ailleurs pas seulement

---

<sup>2</sup> CREST est une mise à jour de REST dans l'optique du web applicatif.

<sup>3</sup> Universal (ou uniform) resource identifier.

<sup>4</sup> Universal (ou uniform) resource locator.

d'obtenir une représentation, mais aussi d'en créer une nouvelle, de déposer une nouvelle version ou de supprimer une représentation.

Troisième concept fondamental, HTML, un langage destiné à représenter, stocker et communiquer la représentation des « pages web ». Il a depuis été complété par d'autres langages utilisant une syntaxe XML pour échanger toutes sortes de données structurées et de documents, l'un des dialectes de XML étant une syntaxe (RDF/XML) de RDF, le langage des données liées et le modèle de graphes au cœur du web sémantique.

Ces trois éléments constituent le cœur de l'architecture du web. On prend la mesure de leur importance à condition de bien comprendre que toute extension actuelle, y compris le web de données, est fondamentalement basée sur les URIs et le protocole HTTP pour identifier le sujet des données échangées (URI) ou transférer ces données (HTTP). Cependant, à la différence d'une vision documentaire dans laquelle les relations sont formées entre des ancres dans des documents hypertextes, les relations dans le web de données sont des liens typés (où les types sont eux-mêmes identifiés par des URIs) entre des ressources de toutes natures (également identifiées par des URIs). En s'appuyant sur les HTTP-URIs pour nommer, sur le protocole HTTP pour transférer des données, sur le modèle de graphe RDF (au lieu de HTML) pour décrire et relier des ressources, et sur des schémas partagés, les recommandations du web sémantique tracent les grandes lignes d'une architecture visant une interconnexion mondiale à la fois des sources de données et de leurs modèles. Cette interconnexion porte virtuellement sur tout ce que l'on se donne la peine d'identifier au moyen d'une URI ; en d'autres termes, tout ce pour quoi l'on crée une ressource.

## 1.2 La crise d'identité du web

Les normes relatives aux identifiants ont évolué au fil du temps. Une première ébauche en fut donnée par ce que l'on appelait des UDIs<sup>5</sup>, des identifiants de « documents » et de « choses » [3]. Puis vint la spécification des URIs [4] avant la naissance du W3C, alors que les fondamentaux du web n'étaient pas encore clairement distingués de leurs implémentations. Il y eut aussi les URNs<sup>6</sup> [29] : noms propres ou identifiants non déréférencables, supposés de ce fait pérennes, formant l'ensemble complémentaire des URLs dans les URIs. Sans parler des URCs<sup>7</sup>, une initiative avortée d'associer des métadonnées descriptives et des protocoles spécifiques d'accès à celles-ci à l'infrastructure d'identification des URNs.

Le travail accompli par Roy Fielding sur les architectures de type REST [13] [14] a joué un rôle important dans le remodelage de la compréhension tant des identifiants du web que du protocole HTTP. C'est également dans la thèse de Fielding que les ressources furent véritablement définies pour la première fois. Conséquence pratique immédiate de REST en 1997-1998, URLs et URNs fusionnèrent, laissant place à des URIs déréférencables, après avoir été scindées trois ans auparavant dans les premiers standards du web qui firent suite à la naissance du W3C en 1994.

L'économiste et sociologue français Laurent Thévenot [31] résume l'action des normes en expliquant qu'elles sont des « formes » qui visent à généraliser, étendre, stabiliser et rendre interopérable (dirait-on aujourd'hui) une réalité technique donnée. Il s'agit très exactement de ce que le web a réalisé à travers REST et les recommandations qu'il a inspirées. Nous parlons cependant bien ici d'un changement de compréhension plus que d'une évolution architecturale puisque toutes les primitives utilisées par REST sont déjà présentes dans la première version du protocole HTTP au début des années 1990.

Vers la même époque (1998-1999), d'autres recommandations pour l'établissement d'un web sémantique firent leur apparition. Le web avait alors atteint une certaine maturité et un débat animait la communauté. Anciennement connu sous le nom barbare de « *HTTP Range 14* » [17] [20] et maintenant désigné par le non moins barbare « *issue 57* » [18], ce problème consistait à comprendre comment il se pourrait établir une distinction entre des URIs qui identifient de prétendus « documents » et celles qui identifient des « choses » en général. Cette distinction a été elle-même reformulée en termes de « ressources informationnelles » (IR<sup>8</sup>) et « ressources non informationnelles » (NIR<sup>9</sup>) sans que l'on s'inquiète véritablement de savoir si ces distinctions entre URIs d'une part et ressources d'autre part étaient congruentes.

Fondamentalement, la question du « *HTTP Range 14* » peut se résumer comme une tentative pour résoudre techniquement la distinction entre IR et NIR en s'appuyant sur l'en-tête HTTP envoyée par un serveur à un client dans une négociation HTTP typique. Il est en fait difficile de discuter la question du « *HTTP Range 14* » à partir d'un point de vue purement technique, car elle est indissociable des conflits d'interprétation qui s'y sont greffés au fil du temps. Ce que le « *HTTP Range 14* » dit en définitive, c'est qu'un en-tête portant le code de réponse 200 sera suivi d'une représentation alors que, s'il porte le code 303, il sera suivi par une URI qui identifie une nouvelle ressource censée donner accès à une représentation. Quant aux réponses utilisant les codes 4XX et 5XX, elles ne donnent accès à rien ni n'indiquent quoi que ce soit de certain.

---

<sup>5</sup> Universal document identifiers.

<sup>6</sup> Uniform resource name, RFC 1737, <http://tools.ietf.org/html/rfc1737>

<sup>7</sup> Uniform resource characteristics, also RFC 1737 and RFC 2483, <http://tools.ietf.org/html/rfc2483>

<sup>8</sup> Informational resource.

<sup>9</sup> Non-informational resource.

**Tableau 1 – Arbitrage rendu par le HTTPRange-14 (cf. [16])**

Code HTTP	Résultat	Indication
200 (OK)	Représentation	Ressource informationnelle (IR) et non informationnelle (NIR) ?
303 (voir autre)	URI	Tout type de ressource notamment des NIR
4XX, 5XX (erreur)	Message d'erreur	Aucune déduction possible

On pourrait déduire du tableau 1 que la question du « *HTTP Range 14* » échoue à établir quoi que ce soit puisque la deuxième colonne ne contient que des représentations et des URIs dans le cas de la redirection tandis que la troisième colonne, indiquant ce qui peut être déduit de la précédente, est laissée ouverte à l'interprétation. Si les ressources sont juste des « ombres » ou des « concepts » [14], alors les ressources informationnelles et non informationnelles ne peuvent être distinguées en fonction de leur accessibilité potentielle : seules les représentations sont en effet accessibles (colonne « résultat »), nullement les ressources (colonne « indication »).

Toutefois, cela ne semble pas constituer l'interprétation par défaut. Et ce alors même que le « *HTTP Range 14* » était censé fournir une séparation claire et définitive entre ressources informationnelles et non informationnelles. La solution technique préconisée n'ayant pas réussi à atteindre cet objectif, une lecture cette fois-ci *normative* des en-têtes de réponse fut promue à sa suite. Ainsi, chaque fois qu'un en-tête 200 est servi, ce que nous obtenons est une IR et les NIR ne doivent par conséquent être servies qu'indirectement, à travers un en-tête 303, en redirigeant vers une IR dont les représentations sont accessibles. Le débat, en définitive, a largement porté sur la pertinence de cette interprétation, principalement motivée par le besoin de déterminer si une URI identifie un document ou une « chose ». En conformité avec les principes fondamentaux de l'architecture du web, notre réponse est que, dans les deux cas, elle identifie une ressource.

De simple relation technique, la redirection est ainsi devenue une bonne pratique préconisée pour la publication de NIR. En dépit du fait que le « *HTTP Range 14* » a complètement échoué en tant que réponse purement technique, une lecture normative demeurerait encore possible. Toutefois, beaucoup sont encore rebutés par la difficulté de mettre en œuvre la redirection ainsi prescrite sur une large échelle, du fait également du changement imposé à la sémantique usuelle de la redirection, dans des cas fréquemment rencontrés tels que les migrations. C'est pourquoi une nouvelle question a récemment été ouverte par le TAG<sup>10</sup>, le groupe en charge d'examiner les fondamentaux de l'architecture technique du web : « *At their meeting in 16th July 2007 the TAG resolved to create a new issue, HttpRedirections-57, as a response to a community request that we give further consideration to the use of the HTTP 303 status codes \*and\* other possible mechanisms of obtaining a description of a resource (typically a non-information resource) where the referenced resource is not capable of providing representations of itself.* [18] »

Se faisant l'écho du rapport du groupe de travail AWWSW<sup>11</sup>, la question porte en réalité davantage sur la définition des URIs, en particulier dans le contexte de RDF : « *When a URI appears in an RDF statement, how can the reader of that statement determine the author's intended meaning? What RDF triples characterize that meaning? Where does the meaning come from? How should the meaning be determined, particularly in the context of the HTTP protocol, for an HTTP URI? Can we codify a suite of nose-following methods for semantic Web use – a recipe one can follow in order to obtain a canonical graph (or "definition", "description resource", "URI documentation") for a URI?* [1] »

Plutôt que de suivre cette piste et de rechercher des moyens supplémentaires de rendre palpable le « sens » d'une URI, nous chercherons ici à donner un sens au web tel qu'il existe et fonctionne actuellement en montrant au contraire sa cohérence fondamentale, intégrant à la fois le web de documents et le web des applications

## 2 *Loc*i de calcul en ligne : de la référence bibliothécaire au *locus* de calcul

Ce que soulignent le travail et la discussion sur REST, c'est que tous les éléments du web que nous découvrons pleinement dans ses dernières évolutions étaient présents dès ses origines. Ce faisant, nous pouvons affirmer que le web n'a jamais été purement documentaire, tout du moins si nous prenons au sérieux les principes fondamentaux de son architecture. Dès lors, nombre des problèmes que nous venons d'évoquer n'ont tout simplement pas lieu d'être.

<sup>10</sup> Technical Architecture Group.

<sup>11</sup> Architecture of the World Wide Semantic Web.

En examinant la définition d'une ressource dans REST, on peut distinguer trois éléments : la ressource elle-même, l'état de la ressource, et la représentation de cet état. Nous allons successivement examiner chacun de ces trois éléments.

## 2.1 Les ressources

Selon la RFC 2396 [6], une ressource peut être n'importe quoi. Roy Fielding l'a appelée une « ombre » ou un « concept », posant de ce fait un abîme entre les ressources et les documents, fussent-ils numériques, c'est-à-dire, en fin de compte, des ensembles binaires de 1 et de 0 *physiquement* hébergés quelque part. Par définition, les ressources ne peuvent jamais être ni hébergées ni directement accessibles, et ne sont manipulées que par le biais de leurs représentations (voir [14] où une section est très justement intitulée « Manipuler des ombres »).

## 2.2 Les états d'une ressource

Les systèmes adhérant à REST sont dits « *stateless* » (sans états) du fait qu'il n'y a pas de session maintenue sur le serveur. Cependant, bien que les ressources demeurent identiques à elles-mêmes (ou du moins le devraient, étant donné qu'il s'agit là d'une déclaration très largement normative, dont la véracité est quotidiennement contredite), elles n'en donnent pas moins des résultats différents au fil du temps, si l'on s'en tient aux représentations servies pour livrer des informations à leur sujet. On doit donc distinguer une ressource de l'état dans lequel on la trouve lorsqu'on l'interroge. Cela fait écho à la distinction bien connue entre les règles et leurs applications ou le logiciel et son exécution.

Alexandre Monnin [22] a par ailleurs suggéré de comprendre les ressources comme des règles<sup>12</sup>, précisant ainsi l'assimilation opérée par Fielding entre ressources et concepts (il convient en effet de noter que, dans la littérature philosophique, les concepts sont souvent traités comme des règles). Assimiler la ressource à une règle permet de mieux comprendre comment et pourquoi les états sont produits. Fondamentalement, une ressource génère des états : au fil du temps (les pages web évoluent, tout comme le résultat des requêtes aux moteurs de recherche ou les résultats d'applications en général) ou ponctuellement, à travers la négociation de contenu (en abrégé « CONNEG ») qui adapte à chaque clic les représentations que reçoivent nos navigateurs lorsque nous surfons sur la toile (en d'autres termes, lorsque nous invoquons les états d'une ressource identifiée par une URI).

Là où l'on parle de règle, on pourrait aussi parler de son écriture, le code (*codex*), et constater comme Michel Serres la généralisation du code au savoir et à l'action en général, notamment au travers de la théorie de l'information et des nouvelles technologies. Sur le web ces codes sont à la fois génériques et individués, publics et privés, sujets et objets. En informatique, « J'existe donc je suis un code [28] », notamment sur le web.

Naturellement, certains cas semblent pour le moins étranges. Est-ce que Tim Berners-Lee est une règle ? Bien sûr que non ! Mais une règle / ressource étant un moyen d'identifier Tim Berners-Lee, ce que l'on appelle « Tim Berners-Lee » dépendra toujours de la façon dont on individue ce « sujet ». Il peut s'agir soit du « fondateur du web », du « directeur du W3C », d'un « homme né de X et Y », ou encore de « Tim Berners-Lee », quelle que soit la façon dont on entend ce qui se tient derrière ce nom propre. Finalement, ce sont quatre ressources différentes ou, en d'autres termes, quatre objets différents, quatre manières différentes de se saisir de quelque chose.

Il est particulièrement important d'opérer cette distinction puisque rien n'assure qu'une ressource corresponde exactement à « une chose réelle », dans le monde, simplement parce qu'elle a été publiée sur le web. D'autant plus que l'objectif du web sémantique n'est pas de trouver un moyen de résoudre cette difficulté. Aux ressources ne doivent pas nécessairement correspondre des descriptions véridiques (comment l'établir techniquement ?). En revanche, elles doivent avoir assez de contenu pour spécifier, comme l'expliquent Fielding et Taylor, ce qu'un auteur « a l'intention d'identifier [14] ».

## 2.3 Les états représentationnels d'une ressource

Les états restent abstraits, comme les ressources, et ne sont pas accessibles en tant que tels. Ce qui est accessible, c'est la représentation de l'état d'une ressource, ces données que l'on reçoit en réponse à une requête HTTP. Ces représentations, qui peuvent être nombreuses pour une ressource donnée, sont susceptibles d'être exprimées dans différents formats. Dans ces conditions, le critère opérant la synthèse de ces représentations sera leur relative fidélité vis-à-vis de la ressource pour laquelle elles ont été servies. En d'autres termes, chacune d'entre elles doit être calculable en tant qu'état acceptable (c'est-à-dire applications de la règle, exécution du code) d'une ressource (c'est-à-dire la règle, le code).

Prenons l'exemple d'une ressource définie par « le texte original de *Macbeth* de Shakespeare ». Une représentation de cette ressource qui donnerait une traduction en français du texte de *Macbeth* encodée en HTML ne serait pas une représentation acceptable. Ce cas illustre un fait simple mais important : même le web 1.0 était un web des ressources, la distinction ressource / représentation ayant été envisagée pour rendre compte tant de la négociation de contenu, comme fonctionnalité du protocole HTTP lui-même, que de la

---

<sup>12</sup> Une règle est un « standard de correction » (Hans-Johann Glock, *Dictionnaire Wittgenstein*, Paris Gallimard, 2003, p. 517) en fonction duquel il est possible d'exercer tout type d'activité (notamment, mais sans toutefois s'y limiter, le calcul).

variabilité des résultats de requêtes adressées à des services, loin donc de l'image d'un web purement statique. Un point qui ne s'est pas démenti, bien au contraire, avec l'avènement du web d'applications.

Nous adhérons donc à la définition des ressources proposée par Justin Erenkrantz comme « *loci* de calcul [11] », avec une légère différence toutefois, une telle définition nous apparaissant convenir parfaitement au web dans son ensemble, et non au seul web applicatif. Les écrits de Justin Erenkrantz s'adaptent très bien au cadre général que nous entendons poser, où les ressources sont assimilées à des règles.

Ce point est encore accentué au regard de la qualification qu'il leur accole. À le suivre, l'une des principales caractéristiques des ressources est d'exhiber une forme de « continuité réticulaire », ce qui sous-tend le double aspect de stabilité et de changement qui les caractérise ontologiquement. Là où le document opérait, en tant qu'opérateur de confiance, comme témoignage d'un acte passé (juridique en particulier), d'une réalité extérieure à lui (le document est hétérothétique<sup>13</sup> [23]), la ressource, outre son contenu, se pense dans un rapport de continuité assumée dans la durée vis-à-vis des représentations qu'elle contribue à générer. En somme, les représentations documentarient une ressource qui n'est pas un document, tout en étant elles-mêmes autre chose qu'un document au sens traditionnel du terme : le résultat d'un calcul créateur de continuité car sans cesse recommencé, et construisant de ce fait un axe temporel (ce que masquent les représentations visuelles les plus courantes des réseaux). La ressource opère dès lors une synthèse entre la répétition de cette opération de calcul et les écarts qu'elle engendre dans des limites circonscrites et néanmoins renégociables.

### 3 Un web de ressources computationnelles liées

Il est maintenant communément admis de fixer des numéros de version sur le web : 1.0, 2.0, 3.0, 2, etc., si bien que l'on peut aisément se laisser séduire par cette mise en scène de plusieurs implémentations du web se succédant les unes aux autres. Pareille présentation est clairement trompeuse. En réalité, nous sommes uniquement en train de comprendre et d'exploiter plus efficacement les potentialités que l'architecture de Tim Berners-Lee avait ouvertes dans les années 1990. Plutôt que de caractériser des versions du web, ces notations semblent surtout révélatrices de la compréhension (limitée) que nous en avons eue et que nous continuons d'en avoir. Dès lors, lorsque nous opposons les *ressources documentaires* aux *ressources computationnelles*, il faut comprendre non que les ressources auraient changé de nature, mais plutôt que le changement vient de l'usage que l'on en fait et de la compréhension que l'on en acquiert.

#### 3.1 Ressources documentaires et ressources computationnelles

Par la dénomination de ressource computationnelle, nous faisons référence au concept de raison computationnelle de Bruno Bachimont [2], qui décrit le noème<sup>14</sup> du numérique, en prolongement du noème photographique de Roland Barthes<sup>15</sup>, comme le « ça a été manipulé ». Le programme est un calcul autothétique<sup>16</sup>, une manipulation qui transforme un objet numérique en un autre objet numérique. C'est cette nature autothétique qui explique d'ailleurs la grande difficulté rencontrée par le W3C à faire référence à des objets du monde réel par l'intermédiaire des URIs. En d'autres termes, l'objet numérique est un « déjà calculé/manipulé ». Cette instabilité constitutive nous semble en très nette opposition avec la notion de document, qui, du point de vue socio-économique, sert de référent et de support à l'établissement de nombreux processus organisationnels.

Par la distinction entre ressource documentaire et ressource computationnelle, nous cherchons à attirer l'attention sur le fait que la stabilité de ce qui pourrait ressembler à des documents sur le web (les représentations) n'est qu'artificielle et aussi éphémère que les interactions qui les génèrent, car la raison d'être du web est justement d'exploiter au maximum la dimension calculatoire et manipulable du numérique pour accélérer la vitesse de transmission (ou, mieux, de *transfert* – le *T* de l'acronyme REST). « Le concept d'information dont nous faisons aujourd'hui usage est celui d'une industrie de l'information comprise comme un signal d'un message dont on contrôle, par un réseau, les temps et espaces de diffusion, devenant par là une marchandise, et dont la valeur est conditionnée par la vitesse de transmission. [30, p. 124] »

Jusqu'à tout récemment, de nombreuses industries n'étaient guère prêtes à exploiter pleinement le potentiel du web. Cette incapacité était selon toute vraisemblance liée à la méconnaissance, voire au dédain dont faisait

---

<sup>13</sup> « Hétérothétique », par opposition à « autothétique », concept proposé par Bruno Bachimont [2] pour signifier que l'inscription numérique ne porte pas la trace des calculs et procédures qui l'ont engendrée. De ce fait, explique Bachimont, « une telle inscription n'a pas de mémoire ». L'hétérothéticité est au contraire la charge d'extériorité, la mémoire, que conserve le document, source de son aptitude à remplir la fonction de témoin d'un passé révolu.

<sup>14</sup> Concept husserlien, l'objet intentionnel dans le vocabulaire de la phénoménologie. Ce que l'on intentionnalise dans la photo, aux yeux de Barthes, c'est le « ça-a-été » (cf. note 15).

<sup>15</sup> Roland Barthes explique dans *La Chambre claire* (Cahiers du cinéma / Gallimard / Le Seuil, 1980) que, par son processus automatique de captation de la lumière, la photographie intègre de manière constitutive le témoignage d'un événement du passé. C'est ce qu'il appelle, le « ça a été » photographique.

<sup>16</sup> Il ne fait référence qu'à des objets numériques. Comme l'écrit Bachimont, il est « autiste ».

preuve une grande partie des professionnels de l'informatique qui tenait ces technologies pour simplistes, voire « amateurs », préférant souvent le développement de clients lourds au détriment d'applications web. Avec l'arrivée à maturité des technologies d'édition et de développement web, les industriels ont progressivement compris comment exploiter au mieux le numérique pour élargir et accélérer leur processus de publication.

La figure 1 met l'accent sur les différences entre la première vision, plutôt documentaire, des ressources web par rapport à une vision actuelle d'un web de ressources computationnelles. Bien entendu, cette distinction est surtout didactique. Dans la pratique, les choses ne sont pas aussi distinctes. Comme nous l'avons expliqué, l'architecture du web portait depuis son origine toutes les potentialités qui sont exploitées aujourd'hui. Certains utilisateurs avertis avaient déjà commencé à les exploiter, comme les acteurs de la communauté wiki par exemple, alors que d'autres se contentaient d'en utiliser uniquement la partie la plus « simple », à savoir le versant documentaire. Il suffit de mentionner le cas des pages personnelles, des pages jaunes, etc.

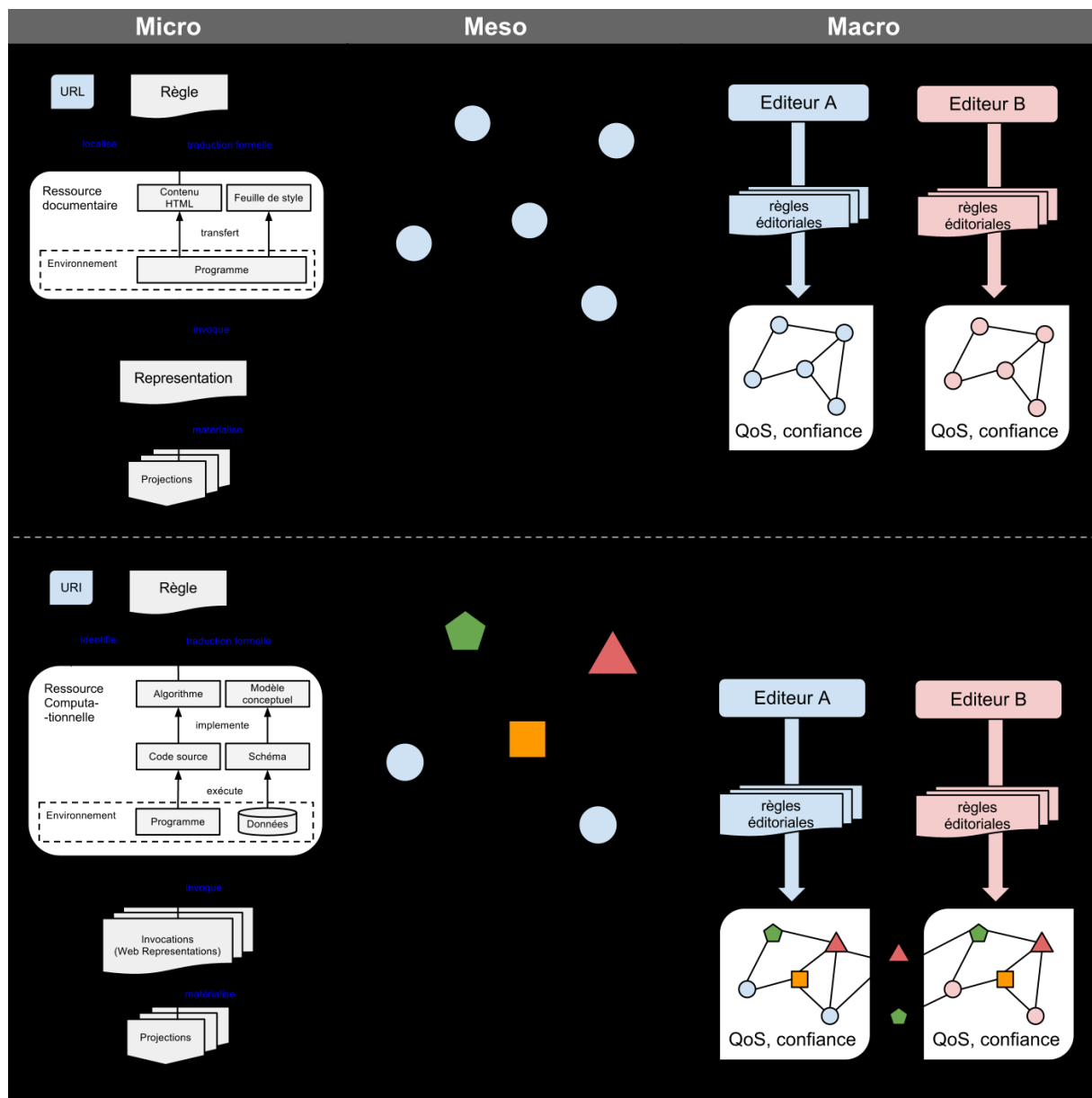


Figure 1 – De l'hypertexte à l'hyperprocess aux niveaux micro, meso et macro

Il est intéressant de jeter un regard sur « l'évolution générale de la compréhension du web ». Cette approche nous permet de souligner combien l'évolution des pratiques a modifié jusqu'à la topologie même du web.

Dans cette section, nous proposons une analyse à trois niveaux de cette « Toile de ressources computationnelles » appelée CoReweb pour *Web of Computational Resources*. Le niveau *micro* se concentre sur la ressource elle-même et ses mécanismes internes. Le niveau *méso* met l'accent sur les relations et interactions immédiates entre les ressources computationnelles. Le niveau *macro* met en évidence les relations entre la politique éditoriale d'un éditeur et la façon dont il gère ses ressources.



### 3.2 Ressources et autres règles

Les ressources du web sont rarement publiées seules et viennent généralement s'intégrer dans un écosystème éditorial plus large, géré par un même éditeur. Nous considérons que la politique éditoriale qui surdétermine le mode d'existence de ces ressources en un tout cohérent peut se concevoir comme un ensemble de règles structurées. Certaines de ces règles sont génériques, d'autres sont spécifiques et peuvent hériter de règles plus génériques, parmi lesquelles on pourra mentionner, à titre d'exemple, une charte graphique commune, des principes de navigation homogènes, des scripts partagés, etc.

De ces prémisses, nous tirons la conclusion que toute ressource web se trouve à l'intersection de  $n$  règles, et se distingue des autres ressources en exprimant formellement une règle spécifique supplémentaire qui la caractérise – par exemple, pour une page web, l'élément qui spécifie son contenu (un article signé par un journaliste, un billet de blog, un message de forum, une opération sur un service, etc.). En d'autres termes, chaque ressource web correspond à  $n+1$  règles, pour peu que l'on associe à une pluralité de règles génériques la règle spécifique qui définit une ressource en propre. Une URI donne donc accès à une représentation qui résulte, de manière ponctuelle, d'un sous-ensemble de l'intersection de ces règles et de sa clôture. Alors qu'elle est souvent perçue comme l'identification d'une seule règle (la plus spécifique des règles impliquées dans la création de la représentation de l'état de la ressource), l'URI donne accès au résultat intégré de l'application de ces règles et de la fermeture transitive de leurs dépendances<sup>17</sup>.

Pour reprendre une analogie avec la linguistique, le « signifié » dans la théorie de Saussure est spécifié par des relations de différence. Les ressources partagent bien des traits communs : elles se lient les unes aux autres, suivent une politique éditoriale donnée, sont organisées en sous-catégories spécifiques, en hiérarchie, etc., ou, alternativement, dans le cas des ressources du web sémantique, elles suivent divers axiomes, partagent des ensembles de propriétés et d'objets, etc. Pourtant, au final, chacune doit avoir un contenu spécifique qui la distingue de ses voisines. Une ressource est précisément cet atome minimum de contenu qui est censé rester stable, du moins autant que faire se peut, alors que les représentations qui leur sont accolées ainsi que les règles correspondant à la politique éditoriale qui les englobe (règles génériques) subissent de fréquentes modifications (sur un rythme cependant plus espacé en ce qui concerne ces dernières). Et c'est là toute la tension entre l'identification d'une ressource au moyen d'une URI, qui se veut pérenne, et l'accès (HTTP-URI) à des représentations générées selon des règles qui tôt ou tard changent.

À cet instant précis, il est utile de mentionner la distinction proposée par T. V. Raman entre les « composants web » et « conteneurs web » : *« The need to provide a single point of access to oft-used information led to portal sites that aggregated all the information onto a single Web page. In this context, the various items of information can be viewed as lightweight Web components. The environment in which these components are hosted (such as the software that generates and manages the Web page) can be viewed as a Web container. Thus, common actions (such as signing in) were refactored to be shared among the various Web applications hosted by the Web container, a piece of software managing the user's browsing context. [24] »*

Les règles susmentionnées reflètent la politique éditoriale d'un « site web », notamment le partage de contenu sur un réseau social, l'utilisation d'un compte enregistré sur un site tiers pour s'inscrire ou se connecter ou encore la possibilité offerte de cliquer sur le bouton Facebook « I like » ou « + 1 » de Google. Ces cas correspondent à l'intégration de composants modulaires (correspondant à l'introduction de nouvelles règles éditoriales), dont l'union donne naissance à un conteneur web. Composants et conteneurs peuvent être ou ne pas être identifiés pour eux-mêmes : un composant Facebook, derrière le fameux « I like », fera l'objet d'une identification au moyen d'une ou plusieurs URIs. En revanche, la décision, ressortissant à la politique éditoriale d'un « site », de lier une page à d'autres pages, sises « à l'intérieur » ou « à l'extérieur » d'un conteneur donné, ne le sera guère.

Dans tous les cas, conteneurs et composants sont comparables à des règles supplémentaires qui viennent ajouter suffisamment de détails au contenu de la ressource pour que l'on soit en mesure de rendre concrètement calculables les représentations HTTP générées à partir de sa « définition spécifique ». Parmi ces règles, mentionnons le logiciel utilisé, les en-têtes du serveur web et sa configuration pour communiquer avec un client, les feuilles de style CSS qui définissent la charte graphique d'un site, le code HTML de la « page », le JavaScript qu'il contient, le JSP ou des balises PHP qu'il utilise, etc. Telles des anneaux de Möbius, ces règles éditoriales sont toujours déjà pourvues de traductions computationnelles sans que l'on puisse isoler l'une ou l'autre face. Elles demeurent non essentielles au sens où elles généreront un effet désiré sans constituer le cœur même de la ressource ; c'est-à-dire sans se confondre avec ce qu'une URI identifie en un sens restreint, soit *spécifiquement*.

Sur le web, maintenir ces politiques éditoriales et les règles qui les définissent est une tâche qui peut être accomplie par une ou plusieurs personnes. Étant donné que ces activités sont souvent réalisées séparément, il est crucial d'être en mesure de les distinguer, ne serait-ce qu'en théorie (les *distinguer* ne revenant nullement à les *isoler* l'une de l'autre).

---

<sup>17</sup> Par fermeture transitive on entend ici l'application de toutes les règles ainsi que récursivement des règles auxquelles elles font référence et des codes dont elles dépendent.

### 3.2.1 Niveau micro

Aux commencements, les navigateurs étaient les seuls logiciels clients connus. À l'heure actuelle, nous disposons d'une multitude d'appareils et d'applications qui sont à tout moment en mesure d'accéder au web et d'interagir avec les services qui y sont proposés.

Les serveurs web ont été initialement conçus pour proposer une sorte de version hypertexte d'un système de fichier distant et distribué. Depuis les interfaces CGI<sup>18</sup>, première manière de standardiser la génération de pages à la volée, leur structure est devenue de plus en plus complexe et les serveurs modernes peuvent maintenant négocier avec les clients pour adapter leur réponse aux divers terminaux. Ainsi, l'immense majorité des contenus du web est générée à la volée, sans être préalablement stockée (dans leur forme finale) où que ce soit. N'importe quel serveur est également compatible avec au moins un langage de programmation (PHP, ASP, C#, Java, etc.) qui peut déclencher, à l'invocation d'une URI (à défaut d'atteindre la ressource directement !), un traitement potentiellement très sophistiqué impliquant d'autres services distants. C'est le cas, par exemple, dans les architectures dites « N-Tiers ».

Ce point a des conséquences importantes lorsque l'on vient à examiner ce qui est téléchargé à partir de ces serveurs. Derrière la vision documentaire des ressources, on peut déceler une tentative visant à préserver la chaîne de causalité entre une référence et un contenu informatif, chaîne constitutive de tous nos systèmes de référence documentaire dans le « monde réel ». La « transition » au plan conceptuel, conduisant à parler de ressources computationnelles et non plus documentaires, souligne à quel point cette relation de cause à effet maintenue artificiellement avait en réalité été brisée d'emblée. Le contenu téléchargé, que Halpin et Presutti [16] nomment une « représentation web » de la ressource, est susceptible de changer à chaque fois qu'une ressource est invoquée. L'emplacement documentaire a été remplacé par le *locus* de calcul d'Erenkrantz – ce que nous appellerions plus volontiers un « espace d'invocations » en perpétuelle renégociation.

Le temps où les « pages » étaient écrites avec des outils de création tels qu'Adobe Dreamweaver ou Microsoft Word semble désormais largement révolu. Aujourd'hui, blogs et wikis ont imprégné la plupart des politiques éditoriales sur le web, alors que les outils et pratiques des origines tendent à disparaître. Du point de vue des serveurs, il est beaucoup plus compliqué d'accueillir un blog qu'un ensemble de fichiers HTML et de feuilles de style CSS. Les outils d'édition en ligne proposent des langages de scripts, ils exigent une prise en charge de la gestion de bases de données ainsi que des politiques de sécurité adéquates pour se connecter à des services distants, ce qui suppose à nouveau la gestion de clés d'authentification à des APIs, etc.

Les URLs, initialement, étaient censées localiser des ressources documentaires. Les interfaces CGI puis les APIs REST ont converti ces URLs en RPC<sup>19</sup> standardisés : des appels distants passant des paramètres aux scripts ou services web répondant aux dits appels.

À l'instar de n'importe quel programme, il convient, pour manipuler une ressource calculatoire, de mettre en œuvre un algorithme associé à un langage de programmation, un modèle conceptuel et des données. Chacune de ces parties a une forte incidence sur les représentations web qu'un utilisateur peut parcourir ou, alternativement, qu'un programme sera habilité à charger. En conséquence de quoi, chaque URL est – et en un sens a toujours été ! – une URI. Les URIs permettent l'identification de ressources protéiformes dont les représentations se peuvent servir dans n'importe quel format requis par un client.

Comme indiqué précédemment, une ressource est une traduction formelle de règles de publication essentielles et non essentielles, mais ces règles elles-mêmes ont tendance à changer au fil du temps (elles ne sont donc, au mieux, que *provisoirement* essentielles !). Leur mise en œuvre évolue pour correspondre à un nouveau contexte technologique. Un bogue sera corrigé, une nouvelle fonctionnalité ajoutée, une base de données mise à jour, etc. Plusieurs raisons expliquent que les représentations des ressources mutent. Et c'est bien là que réside le pouvoir réel de la communication sur le web : un éditeur acquiert la capacité d'adapter instantanément toute la chaîne éditoriale de façon synchrone, en conformité avec des contraintes informationnelles, technologiques, ou tout simplement ponctuelles.

La croissance de la communication sur le web, au cours des quinze dernières années, s'explique surtout par la rapidité avec laquelle un élément d'information peut passer de l'état de données stockées dans une ou plusieurs bases distantes à une représentation web. Ainsi, l'aisance avec laquelle sont mises à jour les chaînes de publication à l'échelle mondiale constitue sa plus grande valeur, en rupture avec les pratiques éditoriales du passé.

### 3.2.2 Niveau meso, l'hyperprocessus

Comme nous l'avons vu, à travers le protocole HTTP, toute ressource computationnelle est susceptible de se référer à d'autres ressources ou de communiquer avec elles. Cette capacité a été exploitée pour ajouter du contenu dynamique en temps réel dans les pages web, mais a aussi pour faire vivre de nombreuses applications du web de données.

---

<sup>18</sup> Common Gateway. <http://www.w3.org/CGI>

<sup>19</sup> Remote procedure call, littéralement « appel de procédure distante ». Cet acronyme sert à désigner toutes les technologies qui permettent d'invoquer l'exécution d'un programme à distance, la plupart du temps à travers le web. SOAP est une technologie de RPC en XML qui a été très utilisée pour l'établissement des premiers services web.

### • Extension des capacités de communication des pages

En 1995, les applets Java offraient la solution la plus pratique pour charger du contenu de façon asynchrone et à distance dans des pages web. Un an plus tard, Microsoft introduisait l'élément *iframe*, conçu pour permettre aux webmasters d'inclure une page dans une autre, ressuscitant au passage le concept de *transclusion* de Ted Nelson (à l'origine du système Xanadu et du mot « hypertexte »). En 1999, le premier contrôle ActiveX, XMLHTTP<sup>20</sup>, est apparu avec Internet Explorer 5. Maintenant, chaque navigateur propose des capacités de communication asynchrones et cette technologie, connue sous le nom d'AJAX<sup>21</sup>, est très largement utilisée.

De nombreux *widgets* (interfaces de composants web) font appel à AJAX pour se connecter à un serveur distant et effectuer en temps réel des changements dans le contenu affiché sur une page: graphiques en temps réel des cotations boursières, bandeau défilant qui affiche les grands titres de l'actualité, Google Maps ou Google Trends sont autant d'exemples d'applications utilisant AJAX.

Mais avec HTML5 et la nouvelle « *JavaScript WebSocket API* », les choses vont encore plus loin. Si AJAX est asynchrone (les connexions sont fermées dès lors que la réponse d'un serveur a été reçue), WebSocket fournit pour sa part des capacités de connexions persistantes entre une page chargée par un client et un serveur. Les connexions persistantes permettent notamment le développement d'applications en temps réel, telles que le dessin collaboratif ou des jeux en 3D. D'autres évolutions comme IndexedDB et WebGL API contribuent encore davantage à transformer des pages web en applications complexes.

Il s'agit là d'une tendance au long cours qui se confirme: de même que les réflexions sur REST offraient au web (et non au seul web de services) une architecture à sa mesure, toute page peut dorénavant se concevoir à la manière d'un service au sens large. Ainsi, par exemple, Florie Bugeaud [9] définit-elle la relation de service comme une « relation de circulation intégrant les entités du service activées en vue de l'effet qu'elles produisent », définition applicable tant aux services web qu'aux simples « pages web ».

### • APIs publiques, tableaux de bord, Widgets, Mashups

Avec la propagation de l'architecture orientée services et la normalisation des protocoles RPC, le web offre un large panel de services en ligne « publics », sur lequel tout développeur peut s'appuyer pour bâtir des applications innovantes. Ces services fournissent souvent des interfaces sous forme de *widgets* (composants), qui permettent de les intégrer très facilement (par copier/coller) dans une page web.

Depuis 2005, de nombreux tableaux de bord (*dashboards*) ont émergé, à l'instar de Netvibes. Très vite, Google, Yahoo! et Microsoft ont publié leurs propres *dashboards*, fournissant une grande variété de composants comme des calendriers, des clients *mail*, des listes de contacts et de *todo-lists*, de lecteurs de flux RSS, d'outils conversion monétaire ou de suivi de la météo.

Des *frameworks* complets, comme LifeRay, ont été développés pour construire ces plates-formes où l'utilisateur peut composer sa propre page faite de composants hétérogènes communément appelés *portlets*. Les applications type tableau de bord semblent actuellement stagner en faveur de *widgets* plus flexibles qui peuvent s'intégrer dans n'importe quelle page. S'il est impossible de les référencer tous ici, nous allons néanmoins en donner quelques exemples parmi les plus typiques :

- *visualisation des données*. En utilisant les protocoles REST ou SOAP<sup>22</sup>, il est désormais courant de composer des chaînes de traitement complexes, faites de multiples appels à des services distants. La combinaison la plus typique est de fournir un flux de données à un service de visualisation et de l'intégrer dans une page web. À titre d'exemple, on peut citer Wordle, ManyEyes ou Google Maps ;
- *mashups*. Un *mashup* est le résultat de la combinaison de plusieurs sources d'information comme des flux RSS. Yahoo! Pipes est une des applications de *mashup* les plus connues et sa cousine, DERI Pipes, inclut des fonctionnalités du web sémantique. Autres exemples : 123people, un agrégateur de renseignements personnels et l'API de Twitter qui a donné naissance à de nombreuses applications comme Bubble-T, Polemic Tweet, etc. ;
- *raccourcisseur d'URLs*. Avec la restriction de Twitter à 140 caractères, les URLs sont souvent trop longues pour être affichées. Pour cela, des services de raccourcissement d'URLs sont apparus comme TinyURL et Bitly. Tous deux fournissent une API publique générant une URL courte à partir d'une première URL plus longue. Ces services sont très simples, intégrés à de nombreux clients Twitter et, bien que la plupart des utilisateurs ignorent parfois qu'ils y font appel, ces services sont parmi les plus utilisés sur le web ;
- *services de traduction*. Pour ceux qui souhaitent obtenir une traduction automatique, dans n'importe quelle langue, du texte de leurs pages web, Microsoft, Yahoo! et Google ont publié leurs solutions sous la forme d'APIs de traduction pour les moteurs Bing, Google (Google Traduction) et autrefois, Yahoo! (Babelfish). La représentation finale visualisée par l'utilisateur est donc l'application de ces services à une représentation web initiale qui pourrait elle-même communiquer avec de nombreuses autres ressources;

<sup>20</sup> XMLHTTP Request, moyen qui permet à un auteur de page web de faire un appel distant à une autre ressource afin de modifier le contenu de la page en cours.

<sup>21</sup> Asynchronous JavaScript and XML.

<sup>22</sup> Technologie d'appel de procédure distante (RPC) basée sur un encodage XML des paramètres.

- la *conversion monétaire*. Il peut être utile de déléguer à un service distant la tâche de conversion de devises en fonction des taux de change courants. Tel est l'objet de services tels que Exchange Rate API ou Open source exchange rates.

#### ● **Orchestration et chorégraphie de services web**

Pour les personnes désirant construire des compositions de services beaucoup plus complexes, afin de modéliser les processus comme des compositions de tâches atomiques et exécuter ces compositions elles-mêmes en tant que processus, plusieurs normes ont été publiées par le W3C, permettant ce qu'on appelle « l'orchestration de services » et la « chorégraphie de services » [19] [27]. On le voit avec cet exemple, les ressources ne sont pas uniquement liées les unes aux autres par des liens de navigation ou de composition. Elles sont également imbriquées dans un réseau d'interactions calculatoires beaucoup plus complexe, constitué d'une multitude de processus enchâssés, exécutés de manière distribuée (par appels RPC), dont le démarrage est déclenché par l'invocation d'une ressource. L'échange de données entre les machines se fait en passant par des formats standardisés comme XML ou JSON.

Au vu de ce qui précède, la qualification du web comme un hypertexte semble bel et bien dépassée ou à tout le moins insuffisante. C'est pourquoi nous lui préférons le terme d'*hyperprocessus*. En fait, REST, en transformant l'interprétation qui prévalait jusqu'alors de l'architecture du web au profit d'une architecture axée sur les ressources désormais promue dans les standards, eut pour effet immédiat de rendre caduque cette notion d'hypertexte – en dépit des formulations de Roy Fielding –, la rendant même totalement inappropriée pour décrire son objet. Malgré sa popularité durable et son importance historique, cette notion est largement dépourvue de sens dans le contexte du web. Au final, on peut aller jusqu'à soutenir que ce dernier aura marqué non l'avènement mais bien plutôt la disparition des hypertextes – souvenons-nous que la proposition inaugurale de Tim Berners-Lee, soumise à la conférence ACM Hypertext de 1991, avait été rejetée sous sa forme originelle pour être finalement présentée en tant que simple poster. L'anecdote est connue, elle demeure édifiante.

#### **3.2.3 Niveau macro, de l'engagement éditorial à l'engagement computationnel**

D'une part, et fort heureusement pour les utilisateurs du web, la complexité croissante des infrastructures de serveurs a été progressivement externalisée sous la responsabilité d'entreprises spécialisées qui fournissent des services d'hébergement et d'administration à faibles coûts. L'amélioration concomitante des technologies de virtualisation et de suivi a grandement simplifié ces tâches d'administration système.

D'autre part, et c'est là l'autre face d'une même pièce, il est de plus en plus difficile pour les éditeurs d'assurer eux-mêmes une bonne qualité de service tout au long de la chaîne de traitements. La pile technologique et les processus impliqués dans la publication d'une ressource sont devenus si complexes, intriqués et distribués, qu'il est de plus en plus difficile de garantir un engagement éditorial strict. Loin de se cantonner à une dimension strictement documentaire, cet *engagement éditorial* s'est transformé en un *engagement de nature computationnelle*, un engagement sur le calcul et la fiabilité de ses résultats. D'objectifs touchant à la qualité éditoriale, les éditeurs ont dû composer avec de nouvelles nécessités touchant cette fois-ci à la qualité du service délivré. C'est précisément ce que Bernard Stiegler décrivait comme une conséquence de la convergence du numérique et de la révolution informatique au sein des industries culturelles [30].

Du point de vue sociétal, les éditeurs de contenu, dont l'activité principale était de produire du contenu et de garantir la qualité de l'information, doivent maintenant faire face à diverses contraintes nouvelles en raison des spécificités d'une plate-forme (support ? médium ?) telle que le web. Au-delà de l'augmentation du rythme de publication, les éditeurs doivent également faire face à de nouvelles attentes du public, du point de vue de la qualité du service technique rendu, en termes d'interopérabilité, ou encore, nous venons de le voir, de pérennité de l'archivage. Les *règles* qui régissaient une ligne éditoriale ont ainsi changé de manière à intégrer, avec plus ou moins de difficultés, les contraintes liées à un dispositif technique.

Désormais, il est parfaitement vain de chercher à dissocier l'*engagement éditorial* de l'*engagement computationnel*. Certains exemples permettent néanmoins de les distinguer *a minima* de façon relativement claire. Ainsi sur Wikipedia, chaque article est doté d'une URI qui l'identifie en tant que ressource. Ressource servie par des représentations éminemment changeantes du fait des principes mêmes de l'édition collaborative d'un wiki. L'engagement de la Fondation Wikimedia est ici double : *computationnel* d'une part, car elle doit proposer un dispositif technique efficace pour que les contributeurs puissent massivement alimenter la plate-forme sans que le dispositif prévu à cet effet ne les décourage, *éditorial* d'autre part, car il est entendu que les articles seront éventuellement modérés, fusionnés, supprimés, renommés, le tout en accord avec une politique éditoriale plus large qui reflète ce que la Fondation veut faire, en terme d'image, de sa plate-forme.

Les éditeurs du web doivent également tenir compte, dans leur engagement computationnel, des règles imposées par les plus gros acteurs comme Facebook, Twitter et Google. Par exemple, la publication doit intégrer les informations permettant un référencement efficace sur les moteurs de recherche. Les articles d'un blog ou d'un journal quotidien doivent intégrer les boutons de Tweeter, « I like » de Facebook ou « +1 » de Google. Les éditeurs qui maintiendraient un niveau d'exigence élevé au niveau de la qualité de la publication, mais qui d'autre part refuseraient de jouer le jeu du web et de s'adapter à ses contraintes, risqueraient de perdre de l'audience en raison d'un décalage entre le service proposé et les attentes du lectorat (d'où une redéfinition profonde de l'acte de lecture lui-même, si tant est que le web s'y cantonne).

Parmi les nombreuses contraintes à prendre en compte dans l'engagement computationnel, et qui pèsent lourdement sur les éditeurs, vient s'ajouter la multiplication des terminaux utilisés pour accéder aux informations : Smartphones, tablettes, télévision connectée, etc. Comme n'importe quel éditeur logiciel, un éditeur de contenu doit concevoir, tester et adapter sa production à chacun de ses supports.

L'informatisation des entreprises étant maintenant quasi achevée, force est de constater que la plupart ont opté pour une externalisation des infrastructures de réseau et des serveurs. Cette externalisation a un coût, non seulement financier mais également en termes de réactivité du fait de l'alourdissement du dispositif technique et humain à mettre en œuvre lors d'une prise de décision. Sans compter que la sous-traitance alourdit d'un intermédiaire supplémentaire la gestion de la qualité de service.

En résumé, l'évolution progressive de l'hypertexte à l'hyperprocessus n'a fait qu'ajouter aux contraintes d'un engagement éditorial celles d'un engagement computationnel. Le second étant au service du premier, cela signifie en réalité qu'il n'y a plus, sur le web, d'engagement éditorial qui ne soit *ipso facto* un engagement computationnel, marquant du même coup une rupture avec les chaînes éditoriales classiques. Cette redistribution des tâches autour de la notion de service n'équivaut cependant nullement à une dématérialisation dont le numérique fournirait le symptôme indubitable. On assiste bien plutôt à une extension inédite du rôle de la technique au cœur même des processus éditoriaux, et ce sur une échelle inédite : la publication, précisément parce qu'elle en sort transformée, n'échoit plus, sur le web, à quelques-uns, mais devient au contraire l'affaire du plus grand nombre.

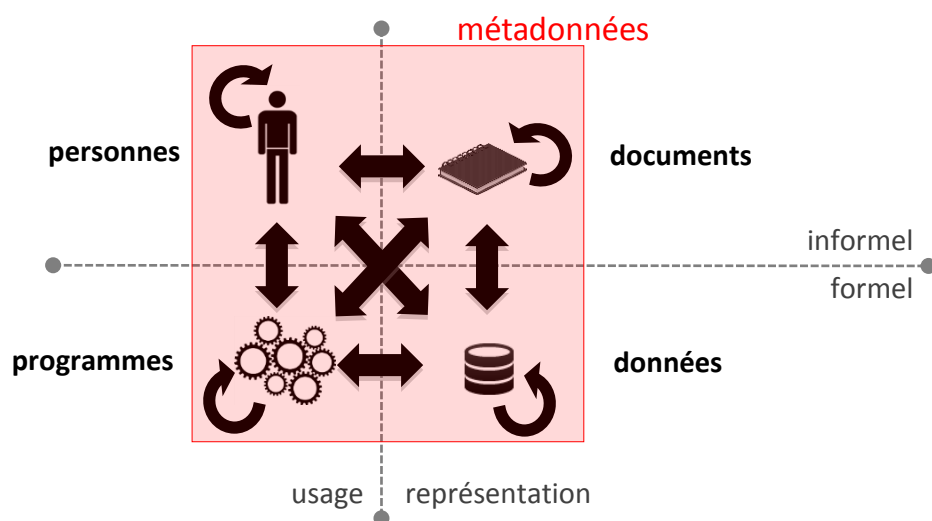
## 4 Vers un web ubiquitaire d'hyper-ressources

La richesse du web n'était guère à démontrer en ce qui concerne la variété des ressources multimédias qu'il organise et met en relation. Cette richesse ne cesse d'ailleurs de croître. Avec l'avènement de l'Internet mobile, comme de l'Internet des objets, nous nous rapprochons à grands pas de la vision d'une réalité augmentée par le web : un web ubiquitaire et un web des objets.

Mais, tandis que le web augmente notre réalité, les objets et les lieux qui nous environnent, nos vies elles-mêmes, ceux-ci viennent en retour augmenter le nombre et la variété des ressources identifiées sur la toile. Cette évolution en forme de cercle vertueux s'accompagne cependant d'un lourd tribut sous la forme d'une augmentation faramineuse de la complexité des ressources et de leur publication. Ceci se fait au risque de rompre avec la relative simplicité qui avait accompagné les premières formes de publication en ligne, le développement des plateformes propriétaires ne fournissant qu'une réponse très partielle à cette difficulté.

L'architecture du web de données et les vocabulaires du web sémantique peuvent à notre sens fournir un moyen de répondre à la diversité des ressources au moyen de métadonnées conçues pour les annoter et utiliser la sémantique de leurs schémas afin de les exploiter plus intelligemment [figure 2]. Les métadonnées et leurs schémas pourraient bien, à terme, constituer la clé de voûte des nouvelles applications centrées sur les ressources, assurant leur intégration et leur interopérabilité. Cependant, si conceptuellement le web sémantique nous entraîne vers des améliorations non négligeables du point de vue des fonctionnalités, il est néanmoins à craindre qu'il alourdisse sensiblement le travail des éditeurs qui, en plus de leurs tâches habituelles, devront prendre en compte ces nouvelles possibilités lors de la publication de leurs données et ressources. Cette évolution ne laisse guère augurer d'un allègement des coûts associés à ce que nous avons appelé l'« engagement computationnel », bien au contraire.

L'enjeu n'en est pas moins là car, demain, quiconque contrôlera les métadonnées contrôlera vraisemblablement les *ressources* et, à travers elles, les *choses*, sur le web comme en dehors. Pour peu que cette distinction elle-même conserve encore un sens à l'avenir...



**Figure 2 – Vue synthétique d'une l'architecture web centrée sur les ressources et unifiée par l'importance transversale des métadonnées (comme dans [15])**

## Références

- [1] ARCHITECTURE OF THE WORLD WIDE SEMANTIC WEB TASK GROUP. *AwwswFinalReport*. 2012. <http://www.w3.org/wiki/AwwswFinalReport>
- [2] Bruno BACHIMONT. *Arts et Sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle*. Mémoire d'HDR, Université de technologie de Compiègne, 2004
- [3] Tim BERNERS-LEE, Jean-François GROFF, Robert CAILLIAU. *Universal Document Identifiers on the Network*. Genève : CERN, February 1992
- [4] Tim BERNERS-LEE. *Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web*. Request for Comments 1630, June 1994. <http://tools.ietf.org/html/rfc1630>
- [5] Tim BERNERS-LEE, Larry MASINTER, Mark McCAHILL. *Uniform Resource Locators (URL)*. Request for Comments 1738, December 1994. <http://tools.ietf.org/html/rfc1738>
- [6] Tim BERNERS-LEE, Roy T. FIELDING, Larry MASINTER. *Uniform Resource Identifiers (URI): Generic Syntax*. Request for Comments 2396, August 1998. <http://tools.ietf.org/html/rfc2396>
- [7] Tim BERNERS-LEE, Robert CAILLIAU, Dan CONNOLLY. *A Little History of the World Wide Web*. 2000. <http://www.w3.org/History.html>
- [8] Tim BERNERS-LEE, Roy T. FIELDING, Larry MASINTER (eds.). *Uniform Resource Identifier (URI): Generic Syntax*. Request for Comments 3986, January 2005. <http://tools.ietf.org/html/rfc3986>
- [9] Florie BUGEAUD. *Isamsara : pour une ingénierie des systèmes de services à base de méréologie et d'hypergraphes*. Thèse, Université de technologie de Troyes, 2011
- [10] Lisa DUSSEULT, James M. SNELL. *PATCH Method for HTTP*. Request for Comments 5789, March 2010. <http://tools.ietf.org/html/rfc5789>
- [11] Justin R. ERENKRANTZ. *Computational REST: A New Model for Decentralized, Internet-Scale Applications*. PhD Thesis, University of California, Irvine, 2009. <http://www.erenkrantz.com/CREST/Erenkrantz-2009-CREST-Dissertation.pdf>
- [12] Roy T. FIELDING, James GETTYS, Jeffrey C. MOGUL, Henrik FRYSTYK, Larry MASINTER, Paul J. LEACH, Tim BERNERS-LEE (eds.). *Hypertext Transfer Protocol – HTTP/1.1*. Request for Comments 2616, June 1999. <http://tools.ietf.org/rfc/rfc2616.txt>
- [13] Roy T. FIELDING. *Architectural Styles and the Design of Network-based Software Architectures*. PhD Thesis, University of California, Irvine, 2000. [http://www.ics.uci.edu/%7Efielding/pubs/dissertation/fielding\\_dissertation.pdf](http://www.ics.uci.edu/%7Efielding/pubs/dissertation/fielding_dissertation.pdf)
- [14] Roy T. FIELDING, Richard N. TAYLOR. « Principled design of the modern Web architecture ». *ACM Transactions on Internet Technology (TOIT)*, 2002, vol. 2, n° 2, p. 115–150. <http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf>
- [15] Fabien GANDON, Catherine FARON-ZUCKER, Olivier CORBY. *Le Web sémantique : comment lier les données et les schémas sur le web ?* Paris : Dunod, 2012
- [16] Harry HALPIN, Valentina PRESUTTI. « The identity of resources on the Web: An ontology for web architecture ». *Applied Ontology*, 2011, vol. 6, n° 3, p. 263–293
- [17] *ISSUE-14: What is the range of the HTTP dereference function?* *HttpRange-14*, [25 March 2002]. <http://www.w3.org/2001/tag/group/track/issues/14>
- [18] *ISSUE-57: Mechanisms for obtaining information about the meaning of a given URI*. *HttpRedirections-57*, [22 August 2007]. <http://www.w3.org/2001/tag/group/track/issues/57>
- [19] Nickolas KAVANTZAS, David BURDETT, Gregory RITZINGER, Tony FLETCHER, Yves LAFON, Charlton BARRETO (eds.). *Web Services Choreography Description Language Version 1.0*. W3C Candidate Recommendation, 9 November 2005. <http://www.w3.org/TR/ws-cdl-10>
- [20] Rhys LEWIS (ed.). *Dereferencing HTTP URIs*. Draft Tag Finding, 31 May 2007. <http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14>
- [21] Alexandre MONNIN. « L'ingénierie philosophique comme design ontologique : retour sur l'émergence de la "ressource" ». *Réel-Virtuel*, 2012, n° 3. <http://reelvirtuel.univ-paris1.fr/index.php?/revue-en-ligne/3-monnin>
- [22] Alexandre MONNIN. « The artifactualization of reference and “substances” on the Web. Why (HTTP) URIs do not (always) refer nor resources hold by themselves ». *American Philosophical Association Newsletter*, 2012, vol. 11, n° 2. [http://hal.inria.fr/docs/00/67/23/01/PDF/Article\\_APA.pdf](http://hal.inria.fr/docs/00/67/23/01/PDF/Article_APA.pdf)

- [23] Alexandre MONNIN. *Préface à : Tom Heath, Christian Bizer. Web de données : méthodes et outils pour les données liées*. Trad. française Julien Plu. Montreuil : Pearson France, 2012.  
<http://www.pearson.fr/livre/?GCOI=27440100179400>
- [24] T. V. RAMAN. « Toward 2W, beyond Web 2.0 ». *Communications of the ACM*, 2009, vol. 52, n° 2, p. 52–59
- [25] T. V. RAMAN, Ashok MALHOTRA (eds.). *Identifying Application State*. TAG Finding, 01 December 2011. <http://www.w3.org/2001/tag/doc/IdentifyingApplicationState>
- [26] Jonathan REES. *The TAG Member's Guide to ISSUE-57 Discussion*. 27 March 2012.  
<http://www.w3.org/2001/tag/2012/04/57guide.html>
- [27] Steve ROSS-TALBOT, Tony FLETCHER (eds.). *Web Services Choreography Description Language: Primer*. W3C Working Draft, 19 June 2006. <http://www.w3.org/TR/2006/WD-ws-cdl-10-primer-20060619>
- [28] Michel SERRES. *Petite Poucette*. Paris : Éditions du Pommier, 2012. 82 p.
- [29] Karen SOLLINS, Larry MASINTER. *RFC 1737 – Functional Requirements for Uniform Resource Names*. December 1994. <http://tools.ietf.org/html/rfc1737>
- [30] Bernard STIEGLER. *La technique et le temps*. Tome 2 : *La désorientation*. Paris : Galilée, 1996
- [31] Laurent THEVENOT. « Les investissements de forme ». In : L. Thévenot, éd. *Conventions économiques*. Paris : Presses universitaires de France, 1986. P. 21-71. (Cahiers de Centre d'étude de l'emploi). <http://gspm.ehess.fr/docannexe.php?id=556>
- [32] Henry S. THOMPSON. *An introduction to naming and reference on the Web*. HTML presented at the Philosophy of the Web seminar, Paris, La Sorbonne, 28 January 2012.  
[http://www.ltg.ed.ac.uk/~ht/Philweb\\_2012](http://www.ltg.ed.ac.uk/~ht/Philweb_2012)